Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2014, Article ID 591920, 15 pages
http://dx.doi.org/10.1155/2014/591920

Research Article

A Regulation-Based Security Evaluation Method for Data Link in Wireless Sensor Network

Claudio S. Malavenda,[1] F. Menichelli,[2] and M. Olivieri[2]

[1] *Selex ES, Via Tiburtina Km 12,400, 00131 Rome, Italy*
[2] *Sapienza University of Rome, Via Eudossiana 18, 00184 Rome, Italy*

Correspondence should be addressed to F. Menichelli; menichelli@diet.uniroma1.it

Received 8 January 2014; Revised 13 June 2014; Accepted 17 June 2014; Published 16 July 2014

Academic Editor: Achour Mostéfaoui

This article presents a novel approach to the analysis of wireless sensor networks (WSN) security, based on the regulations intended for wireless communication devices. Starting from the analysis and classification of attacks, countermeasures, and available protocols, we present the current state on secure communication stacks for embedded systems. The regulation analysis is based on civil EN 50150 and MIL STD-188-220, both applicable to WSN communications. Afterwards, starting from a list of known WSN attacks, we use a correspondence table to match WSN attacks with countermeasures required by regulations. This approach allows us to produce a precise security evaluation and classification methodology for WSN protocols. The results show that current protocols do not present a complete coverage of security issues. While this conclusion is already known for many WSN protocols, to the best of our knowledge this is the first time a complete methodology is proposed to base this assertion. Moreover, by using the proposed methodology, we are able to precisely identify the exposed threats for each WSN protocol under analysis.

1. Introduction

Security assessment of a wireless sensor network (WSN) protocol is a critical issue for a wide range of applications, increasingly pushed by the Internet-of-Things trends. WSNs present peculiar characteristics regarding security feature implementation, because their applications generally address low-energy and battery powered embedded devices, in which the main requirement is to minimize the computational power. Therefore, security attack countermeasures should be implemented with careful attention, due to the limited resources in the system.

This paper aims at identifying a set of measures that should be taken into account while choosing or designing a communication protocol for WSNs, in order to achieve system security and low overhead at the same time. We start with an analysis of the main issues related to security in sensor networks (even if the considerations could be extended to all battery-powered radio devices) where resources are limited and their optimization is a mandatory requirement.

The paper is organized as follows: Section 2 offers an introduction to the main security issues in WSN. Section 3 identifies a WSN attack taxonomy and classifies attacks within OSI layers. Section 4 describes regulations and standards that can be applied to secure wireless sensor networks. In particular, we analyze the rules adopted in the EN 50159 civil standard and in the military MIL STD-188-220D w/Change 1, along with an overview of Common Criteria validation methodology. In particular, Section 4.3 offers an overview of relevant validation criteria for wireless connectivity, especially referring to Common Criteria validation methodology and how it can be applied to WSNs. Section 5 reports the mapping of WSN attacks to countermeasures, and finally in Section 6 we use the proposed methodology to evaluate and classify popular WSN protocols, showing their limitations regarding security issues.

2. Background

Differently from other approaches [1] that classify security issues depending on WSN application, the methodology and results we propose want to address a wide range of common WSN security issues and link them to the lowest

communication layers, that are in common with all protocols and not specific to a single application.

There are three features intended to be addressed while dealing with a secure protocol in WSNs [2]:

(i) *secrecy*: data on the WSN must not be spread to foreign actors;

(ii) *authentication*: a method to identify members authorized to join the network;

(iii) *message replay protection*: a technique to avoid data inconsistency caused by replayed messages.

WSNs are very peculiar from the point of view of security, since they can be subjected to a huge range of vulnerabilities, but at the same time they have few resources to face them. Actually, WSNs are affected by all the known vulnerabilities of a linked network, with the addition of the following:

(i) *vulnerabilities due to the transmission medium*: electromagnetic communications are more accessible than wired ones.

(ii) *vulnerabilities due to unprotected physical access*: a node can be easily captured and should not contain critical information in its program/data memory that could break the whole network.

(iii) *vulnerabilities due to noncentralized network infrastructure*: a centralized authority could be used to certificate authenticity of each node.

(iv) *vulnerabilities due to limited resources*: low computational capabilities, low power available, and limited number of data transmission/reception. This is one of the main constraints for security.

Depending on the nature of the attacker, vulnerabilities can be classified as follows [3]:

(i) *passive*: the attacker listens to the medium and sniffs exchanged packets. This kind of attack is difficult to detect but has no denial of service (DoS) consequences;

(ii) *active*: this kind of attack has the purpose of altering packet routing, gaining authentication, and interacting with the WSN. The attacker can be an external node, that is, a node that does not belong to the network but is capable of interacting with it, or an internal one, that is, an authentic but compromised node. This kind of attack can result in a DoS.

Moreover, the attacker can perform different kind of activities depending on his capabilities that can be classified as follows:

(i) *sensor-class*: the malicious node has computational capabilities similar to a network node;

(ii) *laptop-class*: the malicious node has computational capabilities superior than a network node.

If an attacker can propagate a DoS by compromising other members of the network or moving through the network, the attack is called Dynamic DoS (DDoS). Two main groups of DDoS attacks can be identified [4]:

(i) *mobile*: the attacker can physically move through the network. In this case a malicious node attacks its neighbors, and then it moves to another position and attacks its new neighbors.

(ii) *propagation*: the attacker spreads the center of DoS to infect other nodes. Each infected node affects its neighbors. Variants of this scheme can leave some nodes not infected (e.g., malicious code may spread through the network but it infects a node only after two hops).

Most sensor network protocols feature self-organization of the network and on-demand discovery of the path at routing level; hence there are no problems if one or few nodes cease to exist because of DoS attacks. Things change in case of an increasing number of attacked nodes. Existing WSN solutions completely break down when more than a certain number of nodes are compromised.

End-to-end security mechanisms are generally too computational expensive due to the limited resources available in sensor networks. In [5], for instance, a location-based resilient security approach (LBRS) is proposed through two techniques: location-binding keys and location-based key assignment. In [6] the authors adopt another approach, letting a node communicate not only with his neighbor nodes, but also with nodes at one-hop distance. This protocol allows discovering, if needed, of alternative routing paths and also offers a support for keys management and authentication.

3. Attack and Defense Taxonomy

In this section, we present taxonomy of attacks and common defenses, classified by the communication layer at which they are usually performed [7–9].

3.1. Physical Layer

3.1.1. Jamming. The purpose of this kind of attacks is keeping busy the communication medium used by the network and introducing RF interferences at the same frequency used by the network. In some literature works they are also reported as "collision" attacks related to link layer.

The result of jamming is that packets on air are damaged, or they are never transmitted, if nodes have a collision avoidance mechanism which reports an always busy medium.

A common defense to this attack is the adoption of spread-spectrum techniques for transmission. Namely, there are two different approaches:

(i) FHSS (Frequency-hopping spread spectrum), which is somewhat resistant to an attacker who does not know the hopping scheme, but it may be simply tracked by scanning the transmission.

(ii) DSSS (Direct-sequence spread spectrum), which is more efficient against this kind of attack, since it relies on a wider RF spectrum.

Both types of spread spectrum countermeasures may be neutralized if the attacker is capable of performing a wideband and high-power jamming attack.

3.1.2. Tampering. In this case the attacker gains physical access to a node belonging to the WSN; for example, the node may be physically destroyed or intentionally altered.

Defenses against tampering can be adopted at protocol level (i.e., the communication protocol knows that some node may be destroyed and is designed to keep the network operational even in that occurrence) or at physical level, for example, through geographical separation of nodes, camouflages, or antitampering node packaging. However, these solutions may be expensive and easily bypassed if proper tools are used.

3.2. Link Layer

3.2.1. Exhaustion and Interrogation. This attack is performed by inducing a node to continuously retransmit a packet until the battery of the node is exhausted. This is possible, for example, in protocols that schedule a retransmission when an acknowledgment packet is not received. However, each part of protocol code that schedules a transmission may be attacked and forced to induce exhaustion.

In the IEEE 802.11-based protocols, for example, Request To Send (RTS) and Clear To Send (CTS) packets are used to reserve bandwidth before data transmission. A compromised node could repeatedly send RTS packets in order to elicit CTS packets from a targeted neighbor node, consuming battery power of both nodes. In other protocols, an attacker node sending "path discover request" may induce exhaustion if the malicious node that continuously sends this kind of requests induces the receiving node to respond.

A defense against this type of attack can be implemented by limiting the number of transmissions even to authenticated nodes.

3.3. Network Layer

3.3.1. Hello Flood. Many protocols use an exchange of *Hello* messages to verify neighborhood. This attack exploits routing protocols that require periodic *Hello* packets to be transmitted to announce the presence of a node. In the "hello flood" attack, the attacker broadcasts a high power *Hello* to all nodes and let them think he is a neighbor.

This attack can be avoided by verifying link bidirectionality, assuming that a legitimate node has the same radio capabilities of other WSN nodes. Authentication may be another solution.

3.3.2. Wormhole. Two distinct malicious nodes usually implement this attack. It consists in tunneling packets from a malicious node that sniffs packets to a remote malicious node with long-range radio.

In case an attacker node uses the same hardware platform of the other nodes in the network, a defense against this attack could be implemented by designing a reduced radio range, in order to achieve the required coverage but at the same time avoiding remote replies. A more general countermeasure adopts a geographical routing scheme with timestamps inserted into packets. This measure adds information about node position in each packet that can be used to compute its Time Of Flight (TOF), hence rejecting packets coming from "too distant" nodes.

3.3.3. Sinkhole. In a sinkhole attack, a malicious node attracts traffic from a particular area. All traffic of the area is diverted to the attacker node. The attack is accomplished by compromising the routing algorithm in order to induce source nodes to see a particular node always as "next hop." In that way all traffic will be redirected through the malicious node. If it never forwards packets to neighbors, the attack would be called *black hole*. The black hole [10] attack can also be issued selecting a node with a *homing* attack (see Section 3.3.7) and then physically *tampering* it.

The attack can be avoided using route algorithms resistant to an arbitrary configuration or using a geographical routing.

3.3.4. Sybil Attack. In this attack, the malicious node assumes multiple identities. This kind of attack undermines a distributed solution counting on cooperation. It may be used, for example, to break a cryptographic distributed scheme.

Networks can be defended from the Sybil attack by location verification routines, to be sure that a node is really there where it claims to be, or with proper authentication method to avoid identity fraud typical of this attack [11].

3.3.5. Bogus Routing Info or Misdirection. This class of attacks is focused on altering routing information in several ways, for example, modifying routing packets exchanged between nodes. In on-demand routing protocols that schedule an answer to a path discover request, an attack called *misdirection* can be done with a malicious node replying fake info to the requesting source.

Another kind of bogus routing attack is the *ack spoofing* attack. The malicious node sends false acknowledge messages to each packet-request that sniffs even if it is not the recipient. As a consequence, a weak link can seem strong or a dead node can seem alive even if it is not [12].

Defenses against this kind of attack can be implemented authenticating nodes that send updating route info, while a refresh mechanism may avoid keeping corrupted route data in the network.

3.3.6. Selective Forwarding. A malicious node linked to the network can selectively drop certain packets, usually to favor a route direction. This attack is also known as "*Neglect and greed*." This kind of attack also provides the possibility of altering message priority.

Defenses against this kind of attack can be implemented by using multiple nonjoined routing path obtained using different algorithms.

3.3.7. Homing. This kind of attack is based on theanalysis of traffic patterns in order to identify and target nodes that have special responsibilities. For instance, in a geographic routing scheme, the analysis could allow an attacker to figure out where important nodes are, such as cluster-heads or cryptographic-key managers.

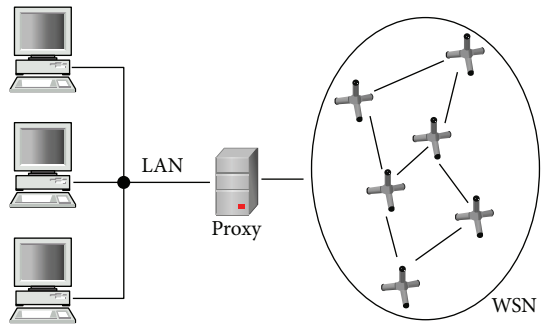


Figure 1: Interconnection of a WSN with remote human interfaces via proxy.

A defense against this attack can be implemented by encrypting headers as well as content [13].

3.4. Transport Layer and Above. Vulnerabilities at transport and higher layers are introduced a wireless sensor network is interfaced with IP-based networks. Up to this point we considered WSN protocols that do not necessarily implement an IP-based communication. Note that from this layer to onwards, the typical security issues of IP-based communications become part of the issues of the WSN architecture; since they are common to IP networks, they will not be addressed in this work.

Figure 1 shows the usual architecture to interface a WSN to the internet, where a proxy linked to an interface node provides access to the WSN. Proxy can be configured in two ways [14]:

(i) as a relay of commands received by remote users toward the WSN;

(ii) as a data storage system, in which the proxy collects data from WSN and stores them in a database, while users perform query on proxy database to read sensor network data.

The limited standardization in the WSN world and the consequent huge variety of protocols let foresee that it is impossible to design general proxy architectures. So, each proxy will have vulnerabilities linked to the specific sensor network it refers to. However a proxy is potentially dangerous for the existence of the WSN itself if configured as a relay, since a malicious user able to gain proxy authentication, could, for instance, flood messages through the WSN and perform an exhaustion attack. On the other hand, a proxy that implements a standard interface for a database is exposed to common database attacks. In both cases, if the proxy fails, for a remote user there is no way to access WSN data.

A particular approach consists in introducing redundancy at proxy level, since it could be the bottleneck of the network, through the adoption of a delay tolerant architecture [15]. In this architecture, several proxies are interconnected and store each incoming/outgoing message, spoofing each other database. No active link is needed among proxies and data is delivered when connectivity is available. A transposition of this concept at LLC layer is analyzed in [16].

4. Relevant Standards in Embedded Wireless Communications

In this section we report the standards that are considered relevant for wireless communications for embedded devices. In particular, we focus on the civil standard [17] EN 50159, with particular emphasis in threats and the military MIL STD - 188-220D w/Change 1 [18] that focuses on a particular implementation of countermeasures. Then we report the standard evaluation criteria used for computer system security, and we identify a complete list of countermeasures applicable to wireless sensor networks, with particular consideration for asynchronous communications.

4.1. EN 50159. EN 50159, also issued as IEC62280, addresses a huge range of security issues for safe communications in railways systems. In particular, "Part1" and "Part2" detail security issues for closed or open, nontrusted transmission systems [17]. The standard does not include particular specifications regarding on which OSI layer should implement these requirements. The protocol architecture could address each security issue at higher layers, that is, at application level, as well as at lower layers.

In WSNs and embedded networks in general, constraints on power consumption and low computational capacity lead to the diffusion of these requirements through the whole protocol definition, in order to optimize available resources.

In EN 50159-2 the following threats are listed:

(1) *repetition*: this include the replication of packets, called also duplication;

(2) *deletion*: a whole packet or just part of it disappears;

(3) *insertion*: an arbitrary formatted message sent by unauthorized node arrives at the receiver;

(4) *incorrect sequence*: in multipacket communications this happens when the receiver is unable to reconstruct the correct message ordering;

(5) *corruption*: message data has been corrupted during transmission;

(6) *delay*: the delay increases over acceptable value;

(7) *masquerade*: this threat happens with authentication errors and also when mixing safety-related messages with non-safety related ones;

Moreover, in the analysis conducted by [19], based over the HAZOP method [20], three threats are added to the above list:

(8) *excessive jitter*: it is the time-displacement of the packet, both in transmission or reception, from its ideal timing;

(9) *inconsistency*: two or more receivers may have inconsistent view of transmitted data or receivers may be in different states when starts receiving the same packet;

(10) *too early messages*: a message may arrive to destination before the receiver is "ready to receive".

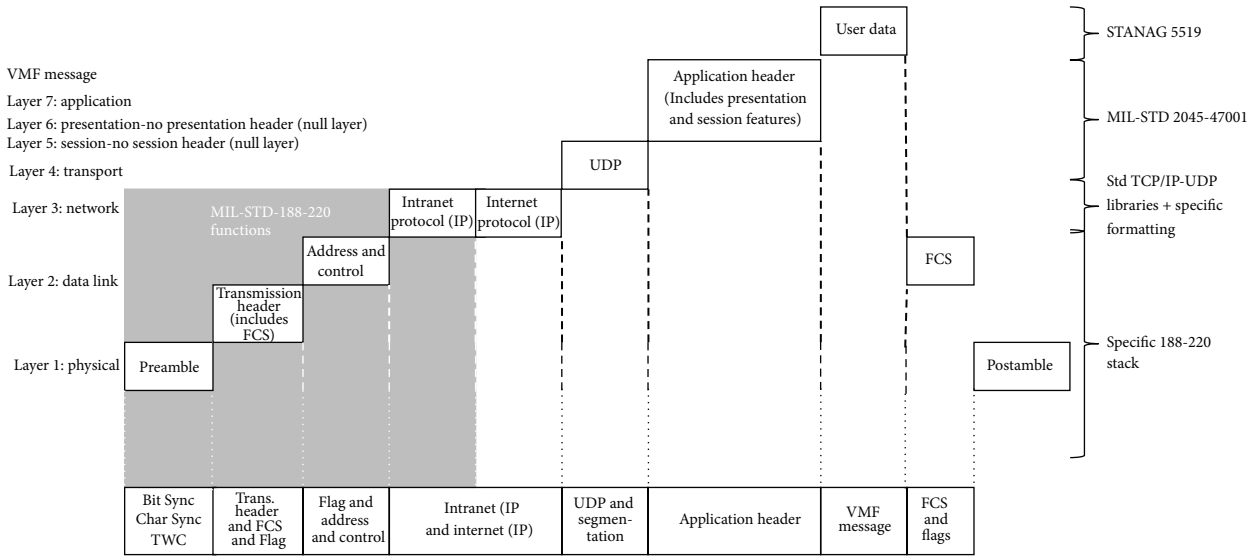


Figure 2: The VMF stack layers.

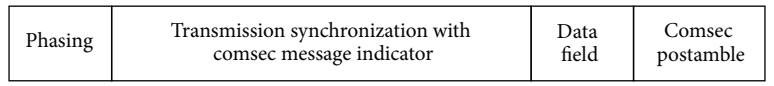


Figure 3: Transmission frame structure with embedded COMSEC.

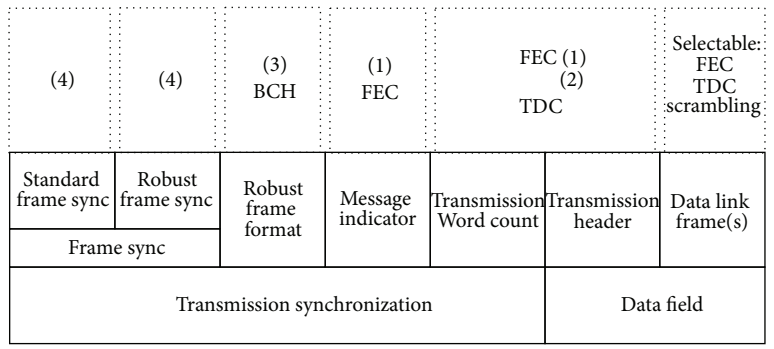


Figure 4: Transmission synchronization field.

4.2. MIL STD-188-220D w/Change 1. MIL STD-188-220 is a military communication standard first released in May 1993, while the latest revision "D w/Change 1" was released in June 2008 [18].

The standard, approved by the US Department of Defense, defines protocol architectures for inter- and intra-communications within Digital Message Transfer Devices (DMTDs) that take part to Command, Control, Communications, Computer, and Intelligence (C^4I) systems. In particular, it defines the lowest communication layers for DMTDs, including wireless embedded devices that take part to the network.

Since it is a military standard, the main focus is on reliability of the protocol. In particular MIL STD-188-220 describes the physical and data-link layers of the communication stack used in wireless devices to exchange VMF messages (Figure 2).

Here we analyze some of the defense mechanism implemented in this protocol, starting from the lowest communication layer, focusing on mechanisms related to asynchronous transmission and on Communication Security (COMSEC) issues, embedded in the communication protocol description.

The transmission frame derived from the standard is reported in Figure 3.

Excluding phasing and postamble fields, used for bit-synchronization and end-of-transmission flagging, our attention will be focused on transmission synchronization and data fields. They are specified as in Figure 4.

The most important fields related to security issues are

(i) *frame sync*: the standard frame sync and robust frame sync are mutually exclusive;

(ii) The *robust frame format*: used only when implementing the *robust communication protocol*, which is

a 7 bit flag that specifies format for the current frame. In particular it refers to the *convolutional coding*, *data scrambling*, and *packetizing* according with the Multidwell Protocol described inside the standard;

(iii) *message indicator* (MI) subfield is a stream of random bits that are redundantly encoded using bit-patterns at physical level. Cryptographic synchronization is achieved when the receiver acquires the correct MI.

(iv) *transmission word count*, which is a 12 bit value calculated by the transmitting station to inform the receiving station of the number of 16 bit words contained in the transmission.

As for the security mechanism, at data-link layer MIL 188-220 adopts the following strategies:

(i) *Golay forward-error-correction* is applied as FEC method. It can be implemented with a feedback shift register, with feedback connections selected according to the coefficients of a polynomial function $g(x)$. It is applied to code the transmission word count, message indicator, and transmission header field;

(ii) *time dispersive coding* is applied on sixteen 24 bit Golay-encoded words ordered to form a matrix. The matrix is then rotated in order to change the bit-sequence. Each block contains a total of 16 FEC codewords. It is applied to the transmission word count and transmission header fields;

(iii) *data scrambling* is applied before FEC coding. It is used when medium has no DC response and the data-sequence to transmit has long NRZ strings (§5.3.15 [18]);

(iv) *convolutional coding* generates three output bits for each input information bit, in order to increase error correction capabilities during reception, using a Viterbi decoding algorithm and hard decisions;

(v) *frame check sequence* is used for error detection. All frames include a 32 bit FCS prior to the closing flag sequence, according to ISO 3309;

(vi) *precedence*: two level-of-precedence bits (L-bits) are used to indicate precedence of data in the information field. The precedence values used in data-link layer can be mapped to precedence values related to network layers with static mapping;

(vii) *Sequence numbers*;

(viii) *network access control* adds robustness introducing functions related to timing. It adds "network busy sensing," "response hold delay," "timeout period," and "network access delay" functions in the MIL 188-220 protocol stack. When using *robust communication protocol* these parameters are modified to achieve greater robustness;

(a) *network busy sensing* is performed at physical level. Its sensing-time can vary according to preamble length;

(b) *response hold delay* is a time that the addressed receiving station waits before sending a response. It is applied during coupled *ack* handshakes. Since its actual value can be calculated only in the receiver station, the source node can only use a prediction of this value. According to the frequency hopping provided by the HAVEQUICK II protocol [21], a time-window is allocated for transmission without interruptions. When ack-messages are used, the ack timing is not known a-priori. This mechanism adjusts the response transmission time and delay parameters in order to meet, where the length of packets allows it, a single hop-window;

(c) *timeout period* starts after the transmission of packets that require an ack. It is a time-window that allows response reception by a source node;

(d) *network access delay (NAD)* is defined as the time that a station with a message which is ready to be sent waits after the time period timer has expired. The standard provides six different schemes to compute this value. Network access delay is always an integer multiple of the Network busy sensing time. All transmissions, except acknowledgments, should begin at the start of the next NAD slot. It could be significantly longer due to extended equipment preamble times, especially for COMSEC operation with HAVEQUICK II hopping protocol;

(e) *multidwell Protocol* defines a series of mechanisms to increase redundancy inside packet bit-stream and packet reconstruction from various source nodes. For example, packet is structured with a "start of packet field" and a "segment counter" with a coded number that stand for the number of 64 bit "data segments" that will be contained in the packet. Transmission is assumed to be one hop, when the receiver detects a "start of packet," it opens a receive time slot. If a frequency-hop is detected during transmission, a handover procedure starts to allow reception from the new transmission frequency. Overhead is added to the bit-stream in order to include Majority choices (2 out of 3 or 3 out of 5) according to the expected BER.

This list of countermeasures adopted by MIL 188-220 will be kept as a reference for the construction of a complete countermeasure list for WSNs.

4.3. Validation Criteria. In this section we expose validation methods used for evaluating computer system security. We aim at finding out methodology guidelines for validation of security countermeasures in WSN systems.

Common criteria (CC) for Information Technology Security Evaluation [22] are international standard coded also ISO/IEC 15408. It offers a framework to evaluate and

certificate security in IT products and systems. It comes out from three preexisting standards: ITSEC [23], CTCPEC [24], and the "orange book" TCSEC [25].

The following paragraphs contain a description of the framework key-points. In the following, the "target of evaluation" (TOE) is the system to be evaluated in its security claims.

Usually a user community or government institution creates a protection profile (PP), that is, a collection of security requirements for a class of IT products. It can be focused on a target user or a particular application of the TOE.

Security target (ST) describes a set of properties of the specific TOE that fulfill the requirements that are expressed in one or more PPs.

Three groups of actors are involved in the security evaluation.

(i) *Consumers*. They express security and protection requirements in an unambiguous manner for an implementation-independent system that can flow in a PP.

(ii) *Developers*. They express security requirements of their implementation in a ST (implementation dependent). This ST may be fully compliant or not with some PPs.

(iii) *Evaluators*. CC describes the set of general actions the evaluators have to carry out, but does not specify procedures.

Evaluation is conducted against countermeasures reported in ST. Two kinds of countermeasures are classified in the framework, countermeasures related to the TOE that must be verified and countermeasures related to the operational framework (safety, operational environment, etc.) that are not verified.

Evaluation of the countermeasures adopted to defend assets from threats goes through two steps.

(i) Evaluation of counter measure sufficiency. The ST describes the adopted countermeasures as security objectives. These are detailed in security functional requirements (SFRs), written in a standardized language, described in Part II of [22]. Evaluation of ST sufficiency is then determined applying to SFRs Security Target Evaluation Criteria, Part III of [22].

(ii) Evaluation of countermeasure correctness. If adopted countermeasures are correct, threats are cancelled. Security assurance requirements (SARs) detail requirements needed to fulfill correctness. Evaluation of correctness is then determined by applying SARs to "evaluation evidences," that is, to a series of tests that can be heterogeneous and various in their nature.

According to the criteria exposed above, our work proposes that a wireless communication protocol should deal with SFRs related to "class FCO communication," that is, "*nonrepudiation of origin*" (FCO_NRO) and "*nonrepudiation of receipt*" (FCO_NRR) that are dependent on "*user identification*" (FIA_UID). In a wireless sensor network the wireless medium and the implemented protocol can be addressed as an "internal communication channel" of the "wireless sensor network" TOE. These enlarge SFRs to "*internal TOE transfer*" (FDP_ITT) and "*Inter-TSF user data integrity transfer protection*" (FDP_UIT).

Table 1: Defense methods versus CC requirements.

SFRs	Counter measure
FCO_NRO	*Source and destination identifier* in each message
FCO_NRR	*Feedback messages* (ack messages)
FIA_UID	*Membership control-bus guardian*
FDP_ITT	*Safety code*
FDP_ACC	*Identification procedure*-address filtering
FDP_IFC	I/O are physically defined in RF circuits
FDP_IFF	Handshake in PtP link
FDP_UIT	*Timeout; timestamp;* shared network identification

The first SFR has dependencies with "*access control policy*" (FDP_ACC) and "*information flow control functions*" (FDP_IFF). This last one inherits itself dependencies from "*Information flow control policy*" (FDP_IFC). An exhaustive description of SFRs is present in [22] part 2.

The complete validation of a protocol with CC is out of the scope of the article, but criteria are taken into account as reference to increase security requirements of a model of WSN TOE and list countermeasures needed to accomplish such requirements. Results are presented in Table 1, where a list of requirements is considered against adopted countermeasures.

This analysis points the light to the need of having an accurate flow control and an active logic on the node that dynamically scans anomalies in received packets.

4.4. Standard Countermeasures List. The analysis of threats conducted in [19] leads to the defense methods reported in Table 2, showing the adoption of a particular defense method against one or more of specific threats. The analysis with respect to the standard validation criteria reported in Section 4.3 points out the completeness of a countermeasure against the threats considered in a regulation.

Threats discussed in this standard are quite general and can be addressed to a wide range of communication protocols, not just wireless ones.

5. WSN Attack Mapping

We can link the considered threats list (Section 4.4) to the attack taxonomy used in WSN (Section 3). Table 3 summarizes this idea. A "X" in the table should be interpreted as "the threat (left column) can cause the attack (upper line)."

Table 3 aims at defining a reference matrix between WSN attack taxonomy and EN50159 threats. It can be taken as reference to introduce the required defense methods in a WSN. For instance, if we want to address the "Sinkhole" threat we find that we need a defense against "Deletion" and "Delay." All related defense method for "Deletion" and "Delay," reported in Table 2, are suitable to contrast "Sinkhole".

Table 2: Description of defense methods against threats.

ID	Defense method	Used against Threat (§IV.A)
A	*Sequence numbers* in each message	1; 2; 3; 4
B	*Time Stamp* of the sending time	1; 4; 6
C	*Timeout* for reception windows	2; 6
D	*Source and destination identifiers* in each message	3
E	*Feedback messages* (ack messages)	3; 7; 9
F	*Identification procedure* about the members identity before a single transmission or at system boot	3; 7
G	*Safety Code* in messages (es.CRC)	5
H	*Cryptographic* techniques	5; 7
I	*Redundancy*, that is, message periodic replication	1; 2; 3; 4; 5
J	*Membership control*: members monitor each other to discover malfunctions. Exception code handling is performed in positive case.	8
K	*Atomic Broadcast*: transmissions are broadcasted to all target nodes in the same order	8
L	*Time-Triggered* architecture: messages are scheduled also with a time-priority handler	1; 2; 4; 5; 8; 6; 10
M	*Bus Guardian*: access to the transmission channel is controlled by a hardware that avoids simultaneous access.	1
N	*Prioritization of messages*: prioritization of messages based on their content-type	8; 6
O	*Inhibit messages*: transmitted messages can't be re-transmitted before a timeout period	1; 6; 8
P	*Hamming Distance* scheme application to node address and message identifiers	3; 7

Particular attention should be mentioned for the tampering attack, as it comes out that tampering is not a common attack covered by regulations. This could be due to the "drop and forget" nature of a WSN node and to the reduced physical dimensions. As from Section 3.1.2, possible countermeasures against tampering attacks are

(i) dynamic routing, which consists in a routing protocol that does not use static routing tables, but can rearrange routing on-the-fly, if required;

(ii) antitamper enclosure, which is a physical countermeasure that aims at detecting physical node manumissions. Since this is a physical countermeasure, not related to the protocol stack, it will not be introduced in Table 2.

Using data in Table 3 along with those expressed in Table 2, it is finally possible to link a full set of countermeasures to WSN attacks taxonomy. Results are presented in Table 4. The number in each cell represents the multiplicity of countermeasures that react to the same kind of attack. Moreover, in Table 4 notation symbols code the defense method nature according to the following list:

(i) ∗—countermeasures related to packet formatting;

(ii) ∗∗—countermeasures that impact handshaking;

(iii) †—countermeasures related to numeric manipulation and transformation of packets;

(iv) ‡—countermeasures that act on physical handling of packets;

(v) #—countermeasure that act on queuing policies;

(vi) ##—countermeasures related to node monitoring and identification;

(vii) $—directives on routing scheme.

This last classification will be used during the layering of each countermeasure, that is, the introduction of a particular countermeasure into a specific communication protocol layer.

6. Security Analysis Applied to Representative WSN Protocols

In this section we apply the correlations reported in Table 4 to analyze the security solutions available in representative and widely known WSN protocols, thus, using the protocols as application examples for the methodology proposed in previous sections. For sake of clarity, we specify that some of them have not been specifically designed to face security risks, nonetheless, due to their popularity, they are analyzed according to our methodology.

We group the protocols taken as case studies according to security features:

(A) intrusion detection and intrusion tolerance protocols: this class refers to features related to detection and tolerance of intrusions;

(B) routing layer protocols: this class refers to routing protocols. A general overview is available in [12];

(C) whole atack protocols: protocols of this class offer countermeasures at several layers.

An overview of each analyzed protocol follows.

6.1. Intrusion Detection and Tolerance Driven Protocols

6.1.1. AODVSTAT. The AODVSTAT [26] protocol introduces a tool into the node to monitor network packets and detect local or distributed attacks within its radio range. The protocol has two modes of operation. In stand-alone mode, a sensor detects attacks within its immediate neighborhood only. In distributed mode, sensors periodically exchange UPDATE messages containing details of near nodes. More precisely, UPDATE messages contain the list of known MAC/IP pairs, the number of hops to known nodes in the network and information regarding detected local attacks.

Table 3: Connection table between WSN attacks taxonomy and EN50159 threats.

Threat	Attack									
	Jamming	*Tampering*	*Exhaustion and Interrogation*	*Selective Forwarding*	*Misdirection*	*Sinkholes*	*Wormholes*	*Sybil Attack*	*Flooding*	*Homing*
Repetition	X		X				X		X	
Deletion				X	X	X	X			X
Insertion	X		X	X	X				X	
Incorrect sequence	X						X			
Corruption	X									
Delay			X	X	X	X			X	X
Masquerade					X			X		
Excessive Jitter	X				X		X			
Inconsistency	X		X	X						
Too early messages	X						X			

This information is then used to detect attacks in a distributed fashion.

AODVSTAT is a sophisticated algorithm for the detection and reaction to a great variety of potential wireless network attacks; unfortunately it is quite computationally expensive.

6.1.2. Effective Intrusion Detection Using Multiple Sensors. This method adopts cooperative and distributed detection algorithms [27] to support intrusion detection, an intelligent routing of intrusion data throughout the network and a lightweight implementation. This is a nonmonolithic system and employs several sensor types that perform specific functions. The control agent is deployed only into a part of network nodes in order to impact performances of only a reduced set of nodes. Data from these sensors are then merged to discover intrusions and try to inhibit the attack.

6.1.3. INSENS (Intrusion Tolerant Routing Protocol for WSN). INSENS [28] constructs forward tables in each node to facilitate communications between sensor nodes and a base station. This protocol does not rely on detecting intrusions, but rather tolerates intrusions bypassing the malicious nodes.

It reduces damages done by intruders by limiting flooding and using appropriate authentication mechanisms, based on symmetric-key cryptography. Moreover it allows alternative network routes to be established between non-malicious nodes. While not providing a real intrusion detection, but rather intrusion tolerance, it still requires a certain number of wireless sensor nodes completely dedicated to this purposes.

6.2. Routing Layer Protocols

6.2.1. AODV. Ad hoc on-demand distance vector is a popular reactive routing algorithm. It is also the routing scheme adopted by ZigBee [29].

When a node of the WSN receives a packet addressed to an unknown destination, it starts sending a "route discovery" packet. This message is flooded through the network until a receiving node finds an entry in its routing table and answers to the request. Each flooded message sets on the receiving device a path in his routing table, since the source of the received message is an in-sight node. In that way, when the message reaches the discovered destination, a backward path is already set.

This kind of protocol introduces a lot of traffic overhead due to path discover flooding. If an attacker log in a relay proxy and is able to generate path discover requests, asking for bogus addresses, a lot of messages overhead will be generated, performing in an easy way an exhaustion attack.

6.2.2. TORA. The temporally ordered routing algorithm (TORA) is an adaptive routing algorithm stacked over the internet MANET encapsulation protocol (IMEP).

TORA logically organizes the WSN nodes as a DAG (directed acyclic graph) using both proactive and reactive path discovering schemes [30].

TORA supports on-demand path discovery very similar to AODV. The protocol uses a particular optimization to perform pro-active routing and adjust the height of the DAG through WSN nodes [31].

This protocol inherits weaknesses of AODV, but limits flooding of "path search" messages with respect to it. Malicious packets spread through the network may compromise the delicate logical organization of a DAG and generate a huge packet overhead to recover the scheme.

6.2.3. DSR. The dynamic source routing protocol (DSR) is a flow oriented routing protocol based on mechanism of route discovering and route maintenance [32]. The route discovering is an AODV-like path discovering mechanism. It is used when the source of message has no entry in its routing table for packet destination. If source node has an entry to route the message, but an intermediate node fails delivering to a next-hop node, route maintenance mechanism is activated. Negative *ack* messages are sent backward to the source and each hop-node which receives that message

Table 4: Correlation between defense methods and WSN attacks.

ID	Defense methods	Attacks										
		Jamming	*Tampering*	*Exhaustion and interrogation*	*Selective forwarding*	*Misdirection*	*Sinkholes*	*Wormholes*	*Sybil Attack*	*Flooding*	*Homing*	
A	*Sequence Number*	3	—	2	2	2	1	3	—	2	1	
B	*Time Stamp*	2	—	2	1	1	1	2	—	2	1	*
D	*Source and destination identifiers*	1	—	1	1	1	—	—	—	1	—	
P	*Hanning Distance*	1	—	1	1	2	—	—	1	1	—	
C	*Timeout*	—	—	1	2	2	2	1	—	1	2	
E	*Feedback messages*	2	—	2	2	1	—	—	1	1	—	**
I	*Redundancy*	4	—	2	2	2	1	3	—	2	1	
O	*Inhibit messages*	2	—	2	1	2	1	2	—	2	1	
G	*Safety Code*	1	—	—	—	—	—	—	—	—	—	†
H	*Cryptographic*	1	—	—	—	1	—	—	1	—	—	
K	*Atomic Broadcast*	1	—	—	—	1	—	1	—	—	—	‡
M	*Bus Guardian*	1	—	1	—	—	—	1	—	1	—	
L	*Time-triggered architecture*	5	—	3	2	3	2	5	—	2	2	#
N	*Prioritization of messages*	1	—	1	1	2	1	1	—	1	1	
F	*Identification procedure*	1	—	1	1	2	—	—	1	1	—	##
J	*Membership control*	1	—	—	—	1	—	1	—	—	—	
Q	*Dynamic Routing*	—	1	—	—	—	—	—	—	—	—	$

Table 5: Minisec performances compared to other protocols.

	Payload (B)	Packet overhead (B)	Security overhead (B)	Total size (B)	Energy (mAs)	Increase over TinyOS
TinyOS	24	12	—	36	0.034	—
TinySec	24	17	5	41	0.0387	13.9%
SNEP	24	20	8	44	0.0415	22.2%
MiniSec	24	15	3	39	0.0368	8.3%

deletes the incorrect entry in its routing table. When the source receives a negative ack, it chooses another routing entry and restarts the delivery process. If no other routing entry is available to the destination then the source starts a path discovering process. MAC layer inherited or DSR-level ack messages may be used according to protocol version. DSR points on reducing flood of messages but is logically susceptible to the same vulnerabilities of AODV.

6.3. Security-Oriented Protocol Stacks

6.3.1. ZigBee. ZigBee defines a security layer based on IEEE 802.15.4 [33]. Three security levels span from no security to encryption of data and security command frames from/to each node.

Particular features of security functions implemented in Zigbee are

(i) it maintains a counter on incoming and outgoing messages to implement reply protection;

(ii) number of integrity bit in a frame can be chosen from 0 to 128;

(iii) authentication and encryption can be chosen between two modes, one key shared from all nodes or one key for each pair of nodes. This last case has a higher memory cost;

(iv) a node can become a "trust center" to provide security keys.

6.3.2. SPINS (Security Protocols for Sensor Networks). This protocol [34] is composed by two linked protocols, SNEP and μTELSA. They can be stacked together to become SPINS or used separately. SNEP and μTELSA will be detailed in the following paragraphs.

6.3.3. SNEP (Secure Network Encryption Protocol). This protocol provides security in node-to-node communications [34]. Main security features it supports are as follows.

(i) *Authentication*. It computes and appends authentication codes to messages. The code is essentially a cryptographically secure checksum.

(ii) *Replay protection*. It is implemented with counters at each node. The checksum code mentioned in the above point is calculated using a secret key and this counter.

(iii) *Semantic security*. It is obtained refreshing the above counter. In fact two identical messages will be encrypted in two different ways if using different sequence numbers.

6.3.4. μTesla. This protocol [34] provides authentication in broadcast communications with low packet-overhead. It support a loosely time synchronization among nodes. Receivers verify that incoming packets have a valid time-signature before storing it. Packets are disclosed immediately after the node receives a key from the packet source node, which periodically distributes them.

6.3.5. TinySec. This protocol implements encryption and authentication. It operates in two modes [35]:

(i) authenticated encryption (TinySec-AE);

(ii) authentication only (TinySec-Auth).

In this protocol packets are partially encrypted, avoiding encrypt/decrypt at each hop.

TinySec explicitly omits replay protection, recommending it to be performed at application layer.

TinySec has been incorporated in TinyOs releases. It requires 728 bytes of RAM and 7146 bytes of program space. The energy overhead imposed by TinySec is 3% for TinySec-Auth and 10% for TinySec-AE.

6.3.6. MiniSec. While TinySec achieves low energy consumption by reducing security countermeasures and ZigBee implements good security features, but suffers for high energy consumption, MiniSec [36] wants to obtain the best of worlds, low energy consumption and high security.

MiniSec has two operating modes, one tailored for single-source communications and another tailored for multisource broadcast communications.

In Table 5, authors of [36] offer a comparison of Minisec against SNEP, Tinysec, and raw transmission with TinyOS to show its low overhead.

6.3.7. ISA100.11.a. The ISA100.11a protocol has been developed by ISA100 committee, formed in 2005. In 2009 it was announced the first release of ISA 100.11.a [37]. The stack is based on the 802.15.4 physical/MAC layers [38] with a 10 ms TDMA scheme. The base version was extended to include the following functions [39]:

(i) link-local addressing;

(ii) message forwarding;

(iii) PHY management;

(iv) adaptive channel-hopping to avoid occupied channels

(v) message addressing, timing and integrity checks;

(vi) detection and recovery of message loss;

(vii) clock synchronization.

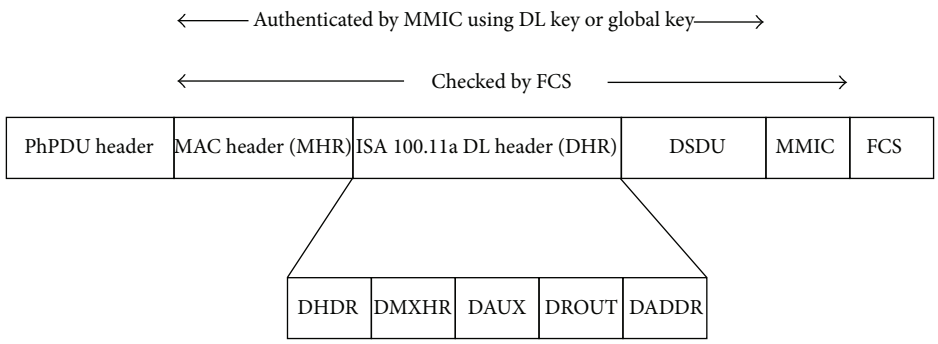
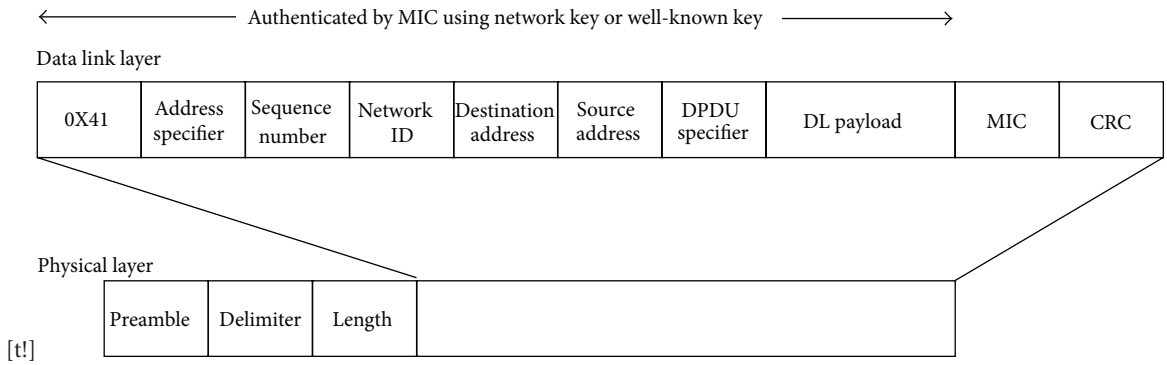


Figure 5: ISA100.11a DPDU.



[t!]

Figure 6: WirelessHART DPDU.

Some of these features can be directly found in the structure of data link layer protocol data unit (DPDU), shown in Figure 5. The routing layer is realized with a graph, several paths may be issued according to network traffic. The transport layer it is based on 6LoWPAN, IPv6, and UDP.

The protocol is designed to be extremely flexible in terms of layers composition, but nodes that implements different version of this standard may not interoperate.

6.3.8. WirelessHART. Introduced into the market in 2007 and approved by the International Electrotechnical Commission (IEC) in 2010 as IEC 62591, the standard is based on the HART "user layer" and the 802.15.4 physical/MAC layers [38]. It introduces its characterization at data link and network layer with several security oriented features that are always turned on and cannot be disabled. In particular, it offers channel hopping support, channel blacklisting (in order to avoid busy frequencies), timing info to support its TDMA implementation (with a fixed 10 ms slot), AES-128 encryption, and dynamic routing (achieved via graphs and time stamps). As stated in [40] WirelessHART implements three security fields:

(i) message integrity code (MIC): this is a sort of CRC code calculated over the whole payload and used during authentication;

(ii) counter: a 4 byte counter used to create the cryptographic *nonce*

(iii) security control byte: a header byte used to define the used security features.

The protocol handles transmission failures through retransmissions using dedicated and shared time slots through different paths in the routing graph [40]. The Network Layer in the WirelessHART protocol stack provides three security services: confidentiality, integrity, and authentication. We can find that some of these features directly match its DPDU structure in Figure 6. WirelessHART and ISA100.11a have several similarities since both are based on 802.15.4 physical/MAC layers and they differ mostly on the layer level where they implement their features [39].

6.4. Protocol Case Study Comparison. Here we present a chart which contains the countermeasures identified with the methodology introduced in Section 5 against all the protocols used as case studies. In Table 6, an "X" means that a particular function/countermeasure is implemented by the protocol.

We can see that all the listed protocols offer some kind of counter measures against security threats. Even protocols not specifically designed for security offer some points of strength. Some of the protocols address just a specific communication need and implement only particular countermeasures, relying on lower layers to integrate security requirements. Usually this approach increases the stack size and resources needed by nodes, since no cross-layer-optimizations can be performed.

Protocols that face "intrusion detection and tolerance" (Section 6.1) usually offer just particular features that let them have only few crosses in Table 6. They are designed to address particular needs; hence they have a limited range of applications. Protocols specifically targeted to routing layer

Table 6: WSN protocols comparison chart.

	AODVSTAT	*Intrusion detection*	*INSENS*	*AODV*	*TORA*	*DSR*	*ZigBee*	*SPINS*	*TinySec*	*MiniSec*	*ISA 100.11a*	*WirelessHART*
Sequence numbers	X	—	—	X	X	X	X	X	—	X	X	X
Time stamp	X	—	—	X	X	X	X	X	—		X	X
Timeout	X	—	—	X	—	—	X	X	—		X	X
S & D identifiers	X	X	X	X	X	X	X	X	X	X	X	X
Ack	X	—	—	X	X	X	X	X	—		X	X
Identification	X	X	X	—	—	—	X	X	X	X	X	X
Safety Code	—	—	—	—	—	—	—	X	—	X	X	X
Crypto	—	—	X	—	—	—	X	X	X	X	X	X
Redundancy	—	—	—	—	—	—	—	—	—	—	X	X
Monitoring	X	X	X	—	—	—	—	—	—	—	—	—
Atomic broadcast	X	—	—	—	X	X	X	—	—	X	X	X
Time-triggered architecture	—	—	—	—	—	—	—	X	—	—	X	X
Bus guardian	—	—	—	—	—	—	X	X	—	—	X	X
Prioritization	X	—	—	X	X	X	X	—	—	—	X	X
Inhibit messages	—	—	X	—	—	—	—	—	—	—	X	X
Hamming distance	—	—	—	—	—	—	—	—	—	—	—	—
Dynamic Routing	X	—	X	X	X	X	X	—	—	—	X	X
		*			‡					$$		

(Section 6.2), namely AODV, TORA, DSR, gain some crosses in the table even if they are not security specifically designed for security issues. Protocols reported in Section 6.3 are an example of stacking, since they offer several countermeasures diffused over many protocol layers. Even if all countermeasures are still not covered, Zigbee and SPINS show good results regarding countermeasures coverage. This is achieved at the expense of stack weight, since few interlayer optimizations are possible. In fact, a complete secure Zigbee protocol stack may exceed 100 kb size of program memory. The most recent standards (ISA100.11a and WirelessHART) achieve the best coverage. They seem to offer a robust and secure stack, introducing most of their countermeasures at data-link layer, while inheriting other layers [41] from interoperable standards. In particular, all security countermeasures are native in WirelessHART and cannot be disabled, remarking how this protocol has been designed to be secure.

Nevertheless, it is possible to see from Table 6 that none of the protocols contains a full implementation of the countermeasures identified in Section 4; however we remark that WirelessHART and ISA100.11a have a full coverage against the threats analyzed even if they are not covering all countermeasures.

These observations lead to the following conclusions:

(1) regulations sometimes require more than one countermeasure against the same threat;

(2) none of the analyzed protocols has a full countermeasure coverage;

(3) the coverage is usually obtained through several firmware stratifications and relying on lower communication layer functionalities, with no interlayer optimization.

We also want to remark that facing all security issues at the lowest communication layers could not even be possible for some embedded platforms that offer a limited access to hardware driver functions.

According to these conclusions, the choice of implementing a communication stack fully customized can introduce the following advantages:

(i) covering all countermeasures;

(ii) offering a lightweight implementation through intralayer optimization and sharing of functionalities.

These main directives should be kept as guideline to design security in protocols dedicated to low-power devices.

7. Conclusions

We presented an overview of general security issues for WSNs and introduced attack taxonomy to catalogue common attacks. The proposed taxonomy is built by analyzing attacks and countermeasures related to each OSI layer involved in the communication, from the physical to the transport layer.

Then we focused on existing standards and in particular the civil EN 50159 and the military MIL STD-188-220D w/Change 1, to find out suitable countermeasures, solutions,

and architectures. In order to validate the countermeasure list, an evaluation of the common criteria regarding wireless protocols was produced. In particular these criteria offer a list of security requirements that a generic "target of evaluation" has to fulfill according to its functionalities.

The complete list of countermeasure was finally mapped in conjunction with the taxonomy introduced for wireless and ad hoc mobile sensor networks. As a result, all identified attacks in WSN have been linked to a countermeasure.

Finally, we applied the methodology to the evaluation of popular protocols for wireless sensor networks. The common characteristic we found is that security functionalities are spread between several layers. This generally affects the protocol by decreasing performances, since security implemented in this way introduces an overhead that can hardly be reduced through optimization and sharing of functionalities.

None of the analyzed protocols covers all security requirements, but the most recent ones seem to cover all threats. In particular, WirelessHART also seems to point out a lightweight solution and a degree of inter-layer optimization.

Future work will leverage on this study to design a new WSN protocol, specific for homeland security and in particular for area monitoring and threats recognition applications, in the view of overcoming all the identified security issues.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

Experiments were conducted in WSN laboratory of Selex ES.

References

[1] R. Singh, D. Singh, and L. Kumar, "A review on security issues in wireless sensor network," *Journal of Information Systems and Communication*, vol. 1, no. 1, pp. 1–7, 2010.

[2] M. Ilyas and I. Mahgoub, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, CRC Press, 2004.

[3] Y. Xiao, *Security in Sensor Networks*, CRC Press, New York, NY, USA, 2006.

[4] J.-S. Lee and Y.-C. Huang, "ITRI ZBnode: a ZigBee/IEEE 802.15.4 platform for wireless sensor networks," in *Proceeding of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1462–1467, Taipei, Taiwan, October 2006.

[5] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "Toward resilient security in wireless sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 34–45, ACM, May 2005.

[6] H. Vogt, "Increasing attack resiliency of wireless ad hoc and sensor networks," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops*, pp. 179–184, June 2005.

[7] C. Gamage, K. Bicakci, B. Crispo, and A. S. Tanenbaum, "Security for the mythical air-dropped sensor network," in *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC '06)*, pp. 41–47, Cagliari, Italy, June 2006.

[8] K. Casey, *Security in Wireless Sensor Networks*, Auburn University, 2005.

[9] D. Wagner, "ecurity for sensor networks: cryptography and beyond," in *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, October 2003, Invited speaker.

[10] P. Mohanty, S. Panigrahi, N. Sarma, and S. S. Satapathy, "Security issues in wireless sensor network data gathering protocols: a survey," *Journal of Theoretical and Applied Information Technology*, vol. 13, no. 1, pp. 14–27, 2010.

[11] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: analysis & defenses," in *Proceeding of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 259–268, April 2004.

[12] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[13] M. Y. Malik, "Outline of security in wireless sensor networks: threats, countermeasures and implementations," in *Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management*, 2013.

[14] A. Dunkels, J. Alonso, T. Voigt, H. Ritter, and J. Schiller, "Connecting wireless sensornets with TCP/IP networks," in *Proceedings of the 2nd International Conference on Wired/Wireless Internet Communications (WWIC '04)*, Frankfurt, Germany, 2004.

[15] F. Warthman, *Delay-Tolerant Networks (DTNs): A Tutorial*, 2003.

[16] C. S. Malavenda, F. Menichelli, and M. Olivieri, "Delay-tolerant, low-power protocols for large security-critical wireless sensor networks," *Journal of Computer Networks and Communications*, vol. 2012, Article ID 863521, 10 pages, 2012.

[17] "EN 50159-2. Railway applications, communication, signaling and processing systems. Part 2: safety-related communication in open transmission system," Brussels, European Committee for Electrotechnical Standardization, 2001.

[18] MILSTD 188-220D_CHG_NOTICE-1, *Digital Message Transfer Device Subsystems*, U.S. Departmnt of Defense, 2008.

[19] J. Alanen, M. Hietikko, and T. Malm, *Safety of Digital Communications in Machines*, VTT, 2004.

[20] IEC 61882, *Hazard and Operability Studies (HAZOP Studies)—Application Guide*, International Electrotechnical Commission, 2011.

[21] HAVEQUICK II, Frequency hopping system, AN/ARC-164, http://mayprinting.com/TSB/data/comm/arc-164.pdf.

[22] CCMB-2012-09-001, "Common Criteria for Information Technology Security Evaluation," v3.1 R4, 2012.

[23] ITSEC, "Information Technology Security Evaluation Criteria," https://www.bsi.bund.de/cae/servlet/contentblob/471346/publicationFile/30220/itsec-en_pdf.pdf.

[24] CTCPEC, "Canadian Trusted Computer Product Evaluation Criteria," Mate Bacic, E., 1990.

[25] TCSEC, *Trusted Computer System Evaluation Criteria*, US Department of Defense Standard, 1985.

[26] G. Vigna, S. Gwalani, K. Srinivasan, E. M. Belding-Royer, and R. A. Kemmerer, "An intrusion detection tool for AODV-based ad hoc wireless networks," in *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC '04)*, pp. 16–27, December 2004.

[27] O. Kachirski and R. Guha, "Effective intrusion detection using multiple sensors in wireless ad hoc networks," in *Proceedings*

of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03), p. 8, IEEE, 2003.

[28] J. Deng, R. Han, and S. Mishra, "INSENS: intrusion-tolerant routing for wireless sensor networks," *Computer Communications*, vol. 29, no. 2, pp. 216–230, 2006.

[29] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, February 1999.

[30] V. Park and S. Corson, "Temporally Ordered Routing Algorithm (TORA) version 1 functional specification, draft-ietf-manet-tora-spec-04.txt," Work in progress. IETF, 2001.

[31] A. A. Pirzada, A. Datta, and C. McDonald, "Trustworthy routing with the TORA protocol," in *Proceedings of the AusCERT Asia Pacific Information Technology Security Conference*, pp. 23–27, 2004.

[32] D. B. Johnson, D. A. Maltz, J. Broch et al., "DSR: the dynamic source routing protocol for multi-hop wireless ad hoc networks," *Ad Hoc Networking*, vol. 5, pp. 139–172, 2001.

[33] ZigBee Alliance, "ZigBee Security Specification Overview," 2005.

[34] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[35] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 162–175, November 2004.

[36] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a secure sensor network communication architecture," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 479–488, April 2007.

[37] "ANSI/ISA-100.11a-2011 Wireless systems for industrial automation: process control and related applications," https://www.isa.org/store/products/product-detail/?productId=118261.

[38] "IEEE Standard for Information Technology Telecommunications and Information Exchange Between Systems—Local and Metropolitan Networks—Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), IEEE 802.15.4," 2004.

[39] Mark Nixon, "A Comparison of WirelessHART and ISA100.11a," Whitepaper, Emerson Process Management, 2012.

[40] S. Raza, A. Slabbert, T. Voigt, and K. Landernas, "Security considerations for the WirelessHART protocol," in *Proceedings of the IEEE Conference on Emerging Technologies & Factory Automation (ETFA '09)*, pp. 1–8, 2009.

[41] T. Lennvall, S. Svensson, and F. Hekland, "A comparison of WirelessHART and ZigBee for industrial applications," in *Proceedings of the 7th IEEE International Workshop on Factory Communication Systems (WFCS '08)*, pp. 85–88, May 2008.